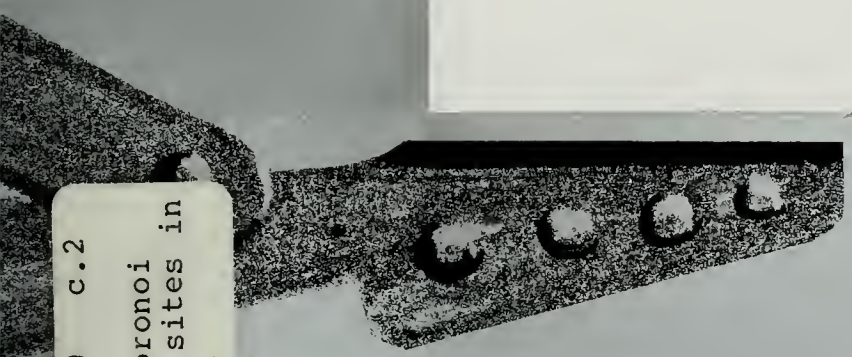
Robotics Research Technical Report

On the Geodesic Voronoi Diagram of Point Sites in a Simple Polygon

by

Boris Aronov

Technical Report No. 390
Robotics Report No. 164
July, 1988



New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012

On the Geodesic Voronoi Diagram of Point Sites in a Simple Polygon

by

Boris Aronov

Technical Report No. 390
Robotics Report No. 164
July, 1988

New York University
Dept. of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

Work on this paper has been supported by Office of Naval Research Grant N00014-87-K-0129 National Science Foundation CER Grant DCR-83-20085, National Science Foundation Grant subcontract CMU-406349-55586, and by grants from the Digital Equipment Corporation and the IBM Corporation.

On the geodesic Voronoi diagram of point sites in a simple polygon.

Boris Aronov

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street, New York, NY 10012

ABSTRACT

Given a simple polygon with n sides in the plane and a set of k point "sites" in its interior or on the boundary, compute the Voronoi diagram of the set of sites using the internal "geodesic" distance inside the polygon as the metric. We describe an $O((n+k)\log(n+k)\log n)$ time algorithm for solving this problem and sketch a faster $O((n+k)\log(n+k))$ algorithm for the case when the set of sites includes all reflex vertices of the polygon in question.

On the geodesic Voronoi diagram of point sites in a simple polygon.

Boris Aronov†

Courant Institute of Mathematical Sciences

1. Introduction.

Recently there has been a significant upsurge of results concerning geometry inside a simple polygon, including an improved triangulation algorithm [TV], efficient algorithms for calculating the geodesic center [PSR] and the geodesic diameter [S2] of a polygon, a number of new shortest-path and visibility-related algorithms that require linear amount of time beyond a triangulation [GHLST], and algorithms for link distance problems [S], [Lea]. Some of the work concentrates on internal distance analogs of fundamental problems for point sets in the Euclidean plane. For example, Toussaint [T] developed an algorithm for computing the "relative convex hull" of a set of points, which is the shortest cycle containing all given points and contained in a given simple polygon, and Suri [S3] described an all-geodesic-furthest-neighbors algorithm for simple polygons. Our present work extends another fundamental problem in computational geometry, namely that of calculating the (nearest-neighbor) Voronoi diagram of a set of point sites (see, for example [PrS]), to the case of points in a simple polygon under the interior geodesic metric (Figure 1).

In sections that follow we present an algorithm which calculates the geodesic Voronoi diagram of a set S of k points ("sites") in a simple polygon with n vertices, using the length of the shortest internal path between two points in the polygon as the measure of distance. Our algorithm runs in time $O((n+k)\log^2(n+k))$, which is not far from being optimal and which is an order of magnitude faster than a previous algorithm of [AA] (see below). Our work can be considered as another generalization of Voronoi diagrams using a new kind of metric. Recent generalizations of similar nature include [A], [AE], [CD], [F], [IIM], [LS], [LW], [L2], [OSY1], [OSY2].

One notable property of the geodesic metric is that calculation of the shortest path between two points is not an easy operation and, without preprocessing, must take $\Omega(n)$ time in the worst case. Also, a single "geodesic bisector" between a pair of sites can be the concatenation of $\Theta(n)$ distinct straight and hyperbolic arcs, which may at first sight suggest that the worst-case overall complexity of the geodesic Voronoi diagram is $\Theta(nk)$. Fortunately, this is not the case, and we show that the total size of the

---

† Work on this paper was performed while the author held AT&T Bell Laboratories PhD Scholarship at New York University.

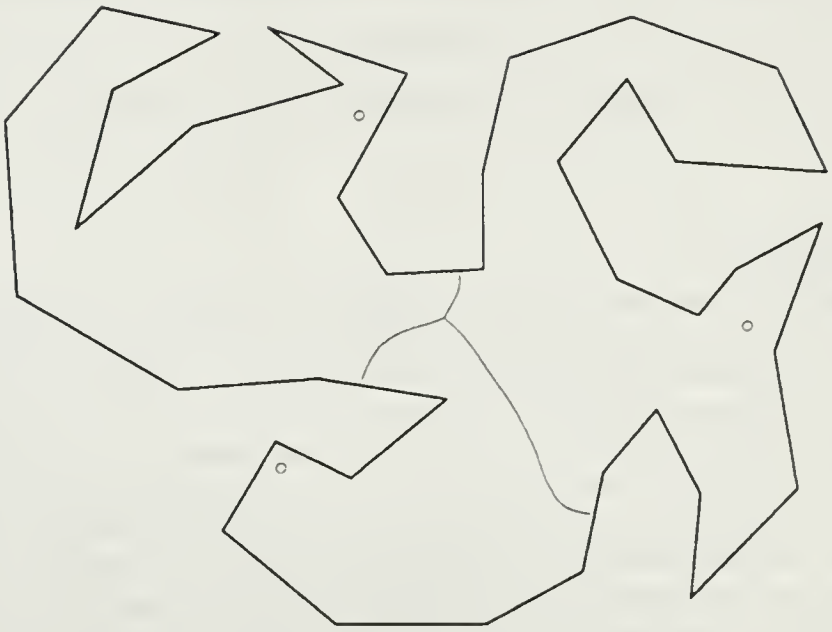


Figure 1.
Geodesic Voronoi diagram of three sites.

diagram is only $\Theta(n+k)$, where only $O(k)$ of the diagram vertices are "true" (degree 3) vertices. Roughly speaking, the diagram can be described by constructing the set of segments that extend the edges of the shortest path tree [GHLST] from each site $s \in S$, but only within the Voronoi cell of s; the augmented diagram can then be regarded as the union of all these sets of extension segments, truncated to their corresponding Voronoi cells and separated from each other by Voronoi edges.

There has been some work done previously in design of an algorithm for constructing the geodesic Voronoi diagram of a set of point sites inside a simple polygon. The best result of which we are currently aware is the algorithm of [AA], which runs in time $O(n\,k + n \log\log n + k \log k)$. The portion of their algorithm that necessitates quadratic time is the explicit construction of the shortest path tree [GHLST] for the *full* polygon from *each* site. As these trees are easily seen to have a total of $n\,k$ distinct edges in the simple case of a convex polygon, the (worst-case) quadratic bound follows. We have circumvented this problem by never building the *full* shortest path tree of the polygon for each of the sites, instead the tree (in fact, a different but closely related "extension segment" structure) from each site is constructed only for the part of the polygon that can conceivably lie in the Voronoi cell of the site.

Our algorithm uses a familiar divide-and-conquer strategy for obtaining the geodesic Voronoi diagram. However, peculiarities of the geodesic metric necessitate a somewhat non-standard implementation of this strategy. The

relatively standard merge step, for example, must be preceded by a step that extends a recursively computed diagram of the subset of sites inside half the polygon to the full polygon. The extension phase is the least conventional part of our algorithm and requires sweeping the triangulation of the polygon by a polygonal scan-line.

Remark: If the set S of sites contains all reflex vertices of P, a simpler algorithm can be used to compute the geodesic Voronoi diagram in time $O((n+k)\log(n+k))$, see discussion in Section 5.

Possible applications of our algorithm include the closest pair problem, the nearest post office problem and all-nearest-neighbors problem in the context of a polygonal universe, such as a (polygonal) island with no interior lakes or a polygonal factory floor. It is likely that other planar point location and proximity problems whose solutions employ Euclidean Voronoi diagram could be generalized to questions about internal metric in a simple polygon and could take advantage of our algorithm. For example, one might wish to investigate how the analogues of a "Delaunay triangulation" and "minimum spanning tree" behave in the context of the geodesic metric.[1]

The paper is organized as follows: Section 2 describes the general structure of our algorithm, Section 3 examines some of the more important geometric properties of the geodesic Voronoi diagram. Section 4 gives a more detailed description of our algorithm. Section 5 outlines a simplified version of the algorithm, which produces the diagram for a set of sites that includes all reflex vertices. Section 6 mentions some related open problems.

2. General outline of the algorithm.

Our approach to solving the problem is the following: Let P be a simple polygon with n sides, and S be a set of k point sites inside P. The *geodesic Voronoi diagram* of S in P, denoted as $Vor_P(S)$, is the partitioning of P into k cells $V_P(s_1), \cdots, V_P(s_k)$ such that

$$V_P(s_i) = \{x \in P \mid \forall j \in \{1, \cdots, k\}\ d_P(x,s_i) \leq d_P(x,s_j)\},$$

where d_P is the geodesic distance inside P. Some initial processing includes triangulation of P [TV] and locating each site of S in the resulting triangles [ST]. Then a balanced decomposition [Ch] of the triangulation tree is computed (see also [GHLST]). The algorithm proceeds by cutting P by a chord into two parts P_1 and P_2 of roughly the same number of edges and recursively computing the diagrams $Vor_{P_1}(S_1)$ and $Vor_{P_2}(S_2)$, where S_1 (S_2) is the subset of S contained in P_1 (P_2, respectively). At this point, the Voronoi diagram $Vor_{P_i}(S_i)$ must be extended to the full polygon P. The two extended

[1] One generalization of Delaunay triangulation (briefly discussed in Section 5) was investigated by Lee and Lin [LL], who defined it directly, and not as the dual of the Voronoi diagram.

diagrams thereby obtained are next merged in a manner similar to the usual Shamos-Hoey scan [SH] (or its modification due to Kirkpatrick [K]). The main novel feature of our algorithm is the diagram extension step, which itself takes $O((n+k)\log(n+k))$ time. The extension is done by propagating, say, $Vor_{P_1}(S_1)$ from the diagonal cut e separating P_1 and P_2 through each of the triangles in P_2 in a preorder traversal of the triangulation (sub)tree of P_2. As the construction proceeds deeper into P_2, two types of events take place: (i) a Voronoi cell of $Vor_P(S_1)$ "dies out," or (ii) a vertex of P_2 is reached, assigned to the Voronoi cell of $Vor_P(S_1)$ containing it, and is incorporated into the shortest path tree within that cell. A "wave front" is maintained and updated at each event. At (i) a Voronoi cell disappears from the wave front and new predictions are made as for the times of disappearance of the two newly adjacent cells. At (ii) the wave front is split in two, one for each of the unvisited triangulation subtrees rooted at the current triangle. Progress is achieved by repeatedly selecting the next closest event to take place and making appropriate changes in the wavefront. A more detailed description can be found in the remainder of the paper. A sketch of a simplified algorithm for the case when S includes all reflex vertices of P is given in Section 5.

3. Properties of the geodesic Voronoi diagram.

One is hard pressed to efficiently compute the geodesic Voronoi diagram before having understood some of its properties and having determined its worst-case complexity. In this section, the definition of the diagram is given, followed by a list of its basic properties and a proof that it has linear complexity.

3.1. Definitions and structural properties.

Let P be a compact region in the plane whose boundary ∂P is a simple n-gon with set of vertices $V=\{v_1, \cdots, v_n\}$. Let $S=\{s_1, \cdots, s_k\} \subset P$ be a set of k point "sites."

Definition 3.1. For any two points v and w of P, let $h(v,w)$ be *the shortest path* between v and w entirely contained in P. The last vertex (or v if there is none) before w on $h(v,w)$ is referred to as the *anchor* of w (with respect to v).

Note 3.2. As P is closed and bounded by a simple polygon, $h(v,w)$ exists and is indeed unique for every pair (v,w) of points of P. Moreover, $h(v,w)$ is piecewise linear with "breakpoints" at vertices of ∂P. For a proof see, for example, [LP].

Definition 3.3. For $v, w \in P$, let *the distance from* v *to* w, $d(v,w)$, be the length of $h(v,w)$.

Note 3.4. Essentially by definition, $d(v,w)$ is a metric on P. Moreover, $d(v,w)$ is continuous in both v and w (with respect to Euclidean metric on

P).

Definition 3.5. The *Voronoi cell* of a site $s \in S$ is

$$V_P(s) = \{x \in P \mid \forall t \in S : d(x,s) \leq d(x,t)\}.$$

Note 3.6. Trivially, $\bigcup_{s \in S} V_P(s) = P$.

The objective of our algorithm is to obtain the decomposition of P by Voronoi cells. Observe that applying known techniques for planar point location to such a decomposition allows efficient computation of $\{ s \mid x \in V_P(s) \}$ for an arbitrary query point $x \in P$.

Definition 3.7. The *shortest path tree* of P from a site s, $T(P,s)$, is the union of the shortest paths from s to vertices of P. It is known that $T(P,s)$ is indeed a planar tree rooted at s with straight-line edges and $V \cup \{s\}$ as the set of vertices (see, for example, [LP]). Assuming s is not a vertex, $T(P,s)$ has exactly n edges and $n+1$ vertices and each of its edges is either a side or an interior chord of P (Figure 2).

Definition 3.8. The *shortest path partition* of P from s is the partition of P into maximal sets each containing points x all having the same anchor with respect to s (thus all the paths $h(s,x)$ pass through the same sequence of vertices of P).

Let e be an edge of $T(P,s)$ and let its endpoint further from s be v. Let r be the open half-line collinear with e and extending from v in the direction of increasing distance from s. If some initial section of r is contained in the

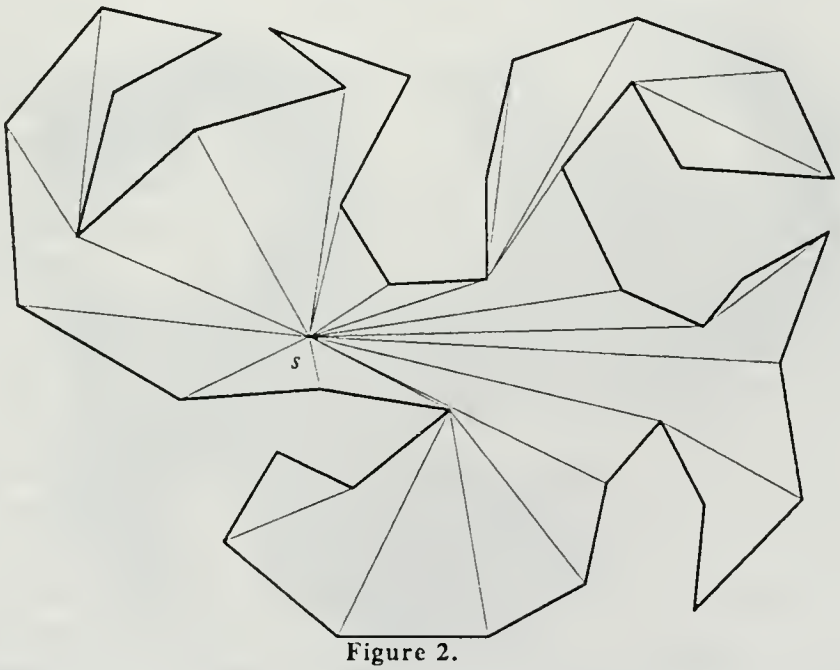


Figure 2.
Shortest path tree $T(P,s)$ of P from s superimposed on P.

interior of P, we will refer to the maximal such initial section as the *extension segment* of e (or the extension segment *emanating from* v). Otherwise e has no extension segment.

Definition 3.9. Let the collection of extension segments of edges of $T(P,s)$ be denoted by $E(P,s)$. We will often abuse the notation and write $E(P,s)$ for the *union* of the extension segments of $T(P,s)$. Note that $E(P,s)$ contains at most n segments—one per vertex of P.

Note 3.10. It was shown in [GHLST] that the partition formed by splitting P along the segments of $T(P,s) \cup E(P,s)$ is a *triangulation* in which points of each triangle share an anchor. Such triangles, however, are not necessarily *maximal* sets with this property. If fact, it is easy to see that splitting P along the segments of $E(P,s)$ alone produces maximal sets and thus corresponds to the shortest path partition of P. In particular, it is a polygonal partition of P (see Figure 3).

Definition 3.11. If $v, w \in P$ and $v \neq w$, we will, following [PSR], define the *direction* from v to w, $\vec{u}(v,w)$, as the unit vector at v directed along the first segment of $h(v,w)$.

Note 3.12. $\vec{u}(v,w)$ is a continuous function of v and w for all pairs of (v,w) *except* when $v = w$ or when there is a segment of $E(P,w)$ emanating from v. Moreover, it is easily verified that $d(v,w)$ is a continuously differentiable function of w with the exception of points noted above and, in fact, $-\vec{u}(w,v)$ is the gradient of $d(v,w)$ with respect to w.

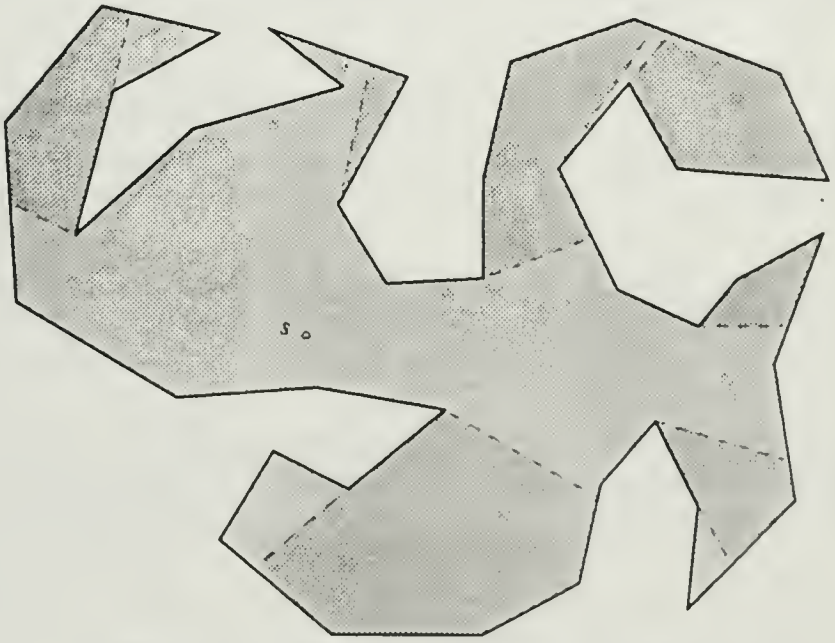


Figure 3. Shortest path partition of P from s.

Definition 3.13. The *bisector* of two (distinct) sites s and t is

$$b(s,t) = \{x \in P \mid d(s,x) = d(t,x)\},$$

i.e., the set of points equidistant from s and t.

Definition 3.14. A set $Q \subset P$ is *star-shaped around* $x \in P$ (with respect to the geodesic metric) if $\forall y \in Q : h(x,y) \subset Q$.

Definition 3.15. If s and t are two distinct sites, let

$$H(s,t) = \{x \in P \mid h(s,x) \cap b(s,t) = \varnothing\}.$$

In other words, $H(s,t)$ is the *maximal* subset of $P - b(s,t)$ star-shaped around s. Note that trivially $s \in H(s,t)$.

Lemma 3.16. $\{H(s,t), b(s,t), H(t,s)\}$ is a partition of P. Moreover, for a point $x \in P$,

$$\begin{aligned} x \in H(s,t) &\Leftrightarrow d(x,s) < d(x,t), \\ x \in b(s,t) &\Leftrightarrow d(x,s) = d(x,t), \text{ and} \\ x \in H(t,s) &\Leftrightarrow d(x,s) > d(x,t). \end{aligned}$$

Proof. It is sufficient to demonstrate the equivalence of our original definition of $H(s,t)$ and the alternative characterization given above.

Assume $x \in H(s,t)$, so $h(s,x) \cap b(s,t) = \varnothing$. We must show that $d(x,s) < d(x,t)$. Continuously parametrize $h(s,x)$ by $\pi : [0,1] \to h(s,x)$ such that $\pi(0) = s$ and $\pi(1) = x$. Note that $d(\pi(\tau),t) - d(\pi(\tau),s)$ is a continuous function of τ on $[0,1]$. It starts off at

$$d(\pi(0),t) - d(\pi(0),s) = d(s,t) > 0.$$

It never reaches zero, as that would indicate an intersection of $h(s,x)$ and $b(s,t)$. Thus it is positive throughout. In particular, $d(\pi(1),t) - d(\pi(1),s) > 0$, i.e., $d(x,t) > d(x,s)$, as desired.

Conversely, suppose $x \notin H(s,t)$, so there exists a point $y \in h(x,s) \cap b(s,t)$. Thus $d(x,s) = d(x,y) + d(y,s)$ and $d(s,y) = d(t,y)$. But then by the triangle inequality

$$d(x,t) \le d(x,y) + d(y,t) = d(x,y) + d(y,s) = d(x,s),$$

as desired, which completes the proof. □

Corollary 3.17. By a simple continuity argument, $b(s,t)$ actually *separates* $H(s,t)$ from $H(t,s)$ in the sense that any path from a point of the former to a point of the latter must intersect $b(s,t)$. In particular, $H(s,t)$ could as well have been defined as the *path-connected* (rather than star-shaped around s) component of $P - b(s,t)$ containing s (Figure 4).

Lemma 3.18. $H(s,t) \cup b(s,t)$ is star-shaped around s.

Proof. Suppose $h(s,x) \not\subset H(s,t) \cup b(s,t)$, i.e., $h(s,x) \cap H(t,s) \neq \varnothing$. Let $y \in h(s,x) \cap H(t,s)$. Then

$$d(s,x) = d(s,y) + d(y,x) > d(t,y) + d(y,x) \ge d(t,x),$$

so $x \in H(t,s)$, as desired. □

Corollary 3.19. $H(s,t) \cup b(s,t)$ is a path-connected and simply connected subset of P.

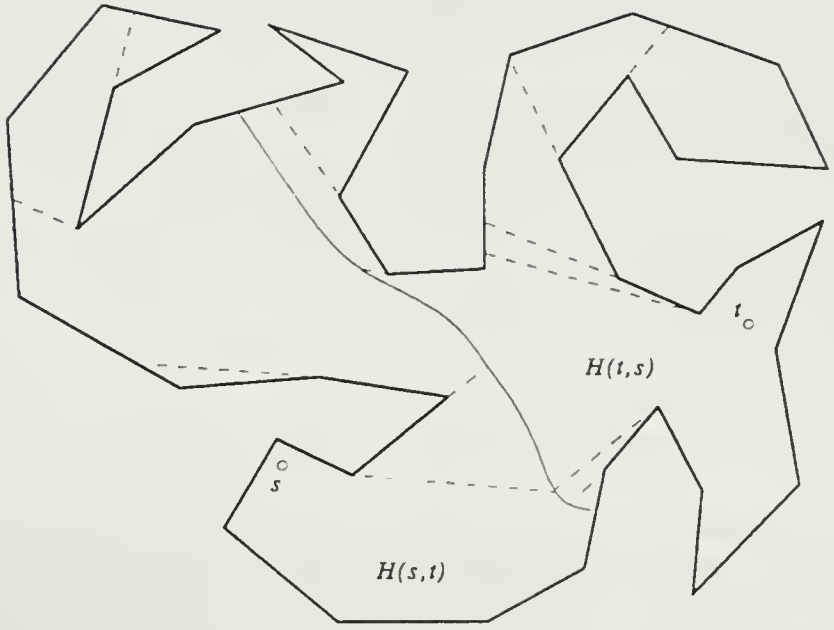


Figure 4.
$H(s,t)$, $b(s,t)$, and $H(t,s)$ together with the shortest path partition.
Solid lines represent $b(s,t)$, dashed – extension segments.

Corollary 3.20. By definition of the Voronoi cell and lemma 3.16, $V_P(s) = \bigcap_{t \in S - \{s\}} (H(s,t) \cup b(s,t))$. In particular, lemma 3.18 implies that $V_P(s)$ is star-shaped around s.

For an alternate proof of the fact that, in general, the Voronoi cell C of a (point) site s is always star-shaped around s in the metric used to define C, see, for example, [PrS].

Definition 3.21. P and S are *in general position* if no vertex of P is equidistant from two distinct sites, i.e., no bisector $b(s,t)$ contains a vertex of P.

Lemma 3.22. Under the assumption of general position, $b(s,t)$ is a smooth curve connecting two points on ∂P and having no other points in common with ∂P. It is the concatenation of $O(n)$ straight and hyperbolic arcs, and the points along $b(s,t)$ where adjacent pairs of these arcs meet are exactly the intersections of $b(s,t)$ with segments of $E(P,s)$ or $E(P,t)$. Moreover, the tangent to $b(s,t)$ at point x bisects the angle between $\vec{u}(x,s)$ and $\vec{u}(x,t)$.

Proof. Partition P into polygonal regions by removing from it $E(P,s) \cup E(P,t)$. Let Q be the closure of one of the resulting regions. By construction there are two *anchor* vertices $\hat{s}$ and $\hat{t}$ such that, for any point x in Q, $d(s,x) = d(s,\hat{s}) + |\hat{s} - x|$ and $d(t,x) = d(t,\hat{t}) + |\hat{t} - x|$, where points are

regarded as vectors and $|\cdots|$ denotes the Euclidean norm. Thus the condition $x \in b(s,t)$ is equivalent in Q to $d(s,\hat{s}) + |\hat{s}-x| = d(t,\hat{t}) + |\hat{t}-x|$. Therefore, in Q, $b(s,t)$ is in general a portion of a hyperbola. The hyperbola in question has $\hat{s}$ and $\hat{t}$ as its two foci and, by elementary analytical geometry, its tangent bisects the angle between the directions to $\hat{s}$ and to $\hat{t}$. The following degenerate cases are possible:

(i) The hyperbola degenerates into a straight line (in fact, the perpendicular bisector of $\hat{s}$ and $\hat{t}$), if $d(s,\hat{s}) = d(t,\hat{t})$ and $\hat{s} \neq \hat{t}$. Note that this is the *only* possibility for a portion of $b(s,t)$ to be a straight-line segment for P and S in general position. In particular, the straight-line portion of $b(s,t)$ *cannot* be collinear with either of the anchors, as this would imply $\hat{s} = t$.

(ii) There are no points equidistant from s and t either because $|d(s,\hat{s}) - d(t,\hat{t})| > d(\hat{s},\hat{t})$ or simply because the hyperbola (or straight line) in question does not intersect Q.

(iii) If $\hat{s} = \hat{t}$ and $d(s,\hat{s}) = d(t,\hat{t})$, *all* points of Q would belong to $b(s,t)$. Then, however, the vertex $\hat{s} = \hat{t}$ would be equidistant from s and t, violating the general position assumption.

(iv) The intersection of the hyperbola with Q consists of discrete points. This situation occurs only when the hyperbola in question intersects the boundary of Q but not its interior.

The above facts imply that $b(s,t)$ can be regarded as the union of (relatively closed) straight and hyperbolic arcs and discrete points. We now proceed to prove a series of properties of $b(s,t)$ which will allow us to conclude that it is indeed a smooth curve connecting two points of ∂P and completely contained in the interior of P otherwise.

(1) Every path-connected component of $b(s,t)$ meets ∂P, for suppose a component C does not—then there is a closed curve p around C that does not intersect either $b(s,t)$ or ∂P. Thus p is completely contained in, say, $H(s,t)$, implying that $b(s,t) \cup H(t,s)$ is not connected—a contradiction.

(2) There are no non-trivial cycles in $b(s,t)$ under the assumption of general position, that is, there is no closed simple curve completely contained in $b(s,t)$. Suppose there exists such a curve and consider a minimal cycle C, i.e., one whose interior contains exactly one connected component of $P - b(s,t)$, say, $H(t,s)$. But then C is a loop in $H(s,t) \cup b(s,t)$ that cannot be contracted to a point, contradicting simple connectedness of $H(s,t) \cup b(s,t)$ (corollary 3.19).

(3) Thus we have proven that $b(s,t)$ is a forest with each tree attached to ∂P in at least one point. We will proceed to prove that all leaves of $b(s,t)$ are points on ∂P. Suppose it is not the case. Let x be a leaf of $b(s,t)$ not on ∂P. Then a sufficiently small open disk D centered at x does not meet ∂P and intersects $b(s,t)$ in a simple curve connecting x to a point on ∂D. Therefore $D - b(s,t)$ is completely contained in either $H(s,t)$ or $H(t,s)$.

Without loss of generality assume that it is contained in $H(s,t)$. As $H(t,s) \cup b(s,t)$ is star-shaped around t, $h(x,t)$ must be contained in $H(t,s) \cup b(s,t)$, and $h(x,t) \cap D \subset (H(t,s) \cup b(s,t)) \cap D = b(s,t) \cap D$, which is impossible in a non-degenerate configuration, as it implies the existence of a straight-line portion of $b(s,t)$ lying on the line passing through one of the anchor points, which was shown in (iii) to contradict the general position assumption.

(4) There are no isolated points of $b(s,t)$ on ∂P, for suppose there were such a point x. Then there is a neighborhood D of x such that $D - \{x\}$ is completely contained in, say, $H(s,t)$ which implies that $H(t,s) \cup b(s,t)$ is *not* connected, contradicting corollary 3.19.

(5) No non-trivial subsegment of ∂P can be contained in $b(s,t)$, as that would imply that both anchors for a point on this segment would have to lie on the straight line containing the segment which, according to (iii), contradicts the general position assumption.

(6) The facts proven so far imply that $b(s,t)$ is a forest all of whose leaves lie on ∂P and which has no non-trivial intersections with ∂P and no isolated points on it. However, corollary 3.17 shows that removal of $b(s,t)$ from P creates *exactly two* connected components. Therefore, the forest in question consists of a single path that does not meet ∂P other than at either of its endpoints, thus proving that $b(s,t)$ is a path connecting two points on ∂P and having no other intersections with ∂P.

Finally, the directions $\vec{u}(x,s)$ and $\vec{u}(x,t)$ vary continuously along $b(s,t)$ (because of note 3.12 and our assumption on general position), so the angle bisection property of the individual hyperbolic arcs also holds at their joining points, implying that the individual arcs are "glued together" smoothly. □

Notice that $b(s,t)$ cannot be tangent to an extension segment of $E(P,s)$, for it would imply (by the previous lemma) that the directions from the point of tangency to s and to t coincide so that $\hat{s} = \hat{t}$, contradicting the general position assumptions (cf. part (iii) of the proof above). In particular, each segment of $E(P,s)$ intersects $b(s,t)$ in at most one point.

Note 3.23. The fact that the bisector of two sites is a smooth curve whose tangent bisects the angle between the directions to the two sites is quite general. Not only does it (obviously) hold for point sites in the Euclidean metric, but it was also shown in [LS] to hold for arbitrary convex sites in the Euclidean plane.

Corollary 3.24. Under the assumption of general position

$$\mathit{interior}(V_P(s)) = \bigcap_{t \in S-\{s\}} H(s,t) = \{x \in P \mid \forall t \in S-\{s\} : d(x,s) < d(x,t)\}$$

and $V_P(s) = \mathit{closure}(\mathit{interior}(V_P(s)))$, where both the interior and the closure are taken relative to P. In particular, $V_P(s)$ is a planar region bounded by sections of ∂P and of bisectors of the form $b(s,t)$, for $t \neq s$.

Corollary 3.25. *Interior* $(V_P(s))$ is star-shaped with respect to s, as each $H(s,t)$ is.

Corollary 3.26. In a non-degenerate configuration,

$$s \neq t \Rightarrow \textit{interior}(V_P(s)) \cap V_P(t) = \varnothing, \text{ for}$$
$$\textit{interior}(V_P(s)) \subset H(s,t) \text{ and } V_P(t) \subset H(t,s) \cup b(s,t).$$

Note 3.27. The previous corollary, which excludes non-trivial overlap among Voronoi cells, is the main reason for restricting our discussion to configurations in general position. If there are two sites equidistant from a single vertex of P, $b(s,t)$ may no longer be a curve, and Voronoi cells (cf. definition 3.5) may overlap non-trivially (see Figure 5).

Lemma 3.28. Let $v \in P$, then $h(s,v) \cap b(s,t)$ consists of at most one point (under the assumption of general position).

Proof. Suppose there are two such points x and y. Without loss of generality assume that y is the point further from s. Since both $H(s,t)$ and $H(s,t) \cup b(s,t)$ are star-shaped with respect to s (definition 3.15 and lemma 3.18) and $h(s,y)$ passes through x, the subpath p of $h(s,y)$ from x to y must be completely contained in $b(s,t)$. By choosing, if necessary, a different x, we may assume p is a straight-line segment and the two anchor points stay constant over p. In particular, $b(s,t)$ contains a straight-line portion p, which forces p to be a part of the perpendicular bisector of the two anchor points. However, p is a portion of $h(s,y)$, and hence is contained in the line passing through one of the anchor points which in turn forces the two anchor points to coincide. However, anchor points cannot coincide on the bisector under the general position assumption by the proof of lemma 3.22. Contradiction. □

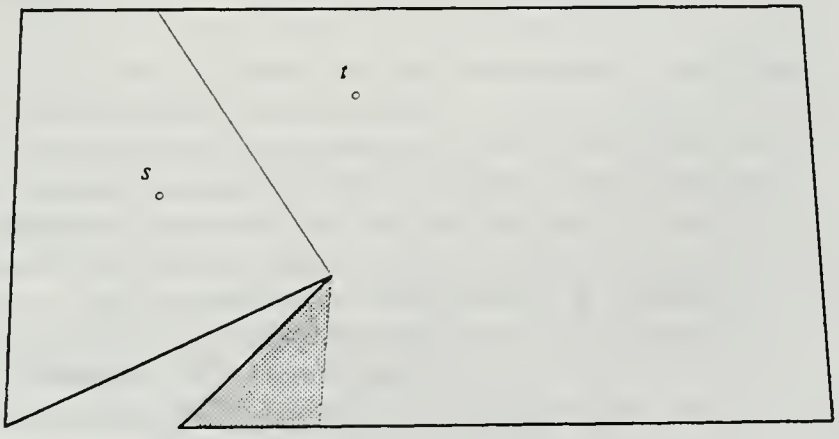


Figure 5.
A configuration not in general position.
Points in the shaded region are equidistant from s and t.

Corollary 3.29. If e is a segment of $E(P,s)$, then $b(s,t) \cap e$ consists of at most one point.

Proof. Apply lemma 3.28, noting that by construction $e \subset h(s,y)$, y being the endpoint of e furthest from s.

3.2. Complexity of the geodesic Voronoi diagram.

Since, by corollary 3.26, Voronoi cell interiors do not overlap, the Voronoi diagram $Vor_P(S)$ of a set of sites S in P is completely described by the union of the Voronoi cell boundaries, the boundary of a cell being the set difference of the cell and its relative interior in P. Note that by corollary 3.26 a point on the boundary of a Voronoi cell necessarily lies in at least two cells and thus is closest to (at least) two sites simultaneously. Consider the planar map on P induced by the union of Voronoi cell boundaries with boundaries taken relative to P. A point in three or more Voronoi cells, or on ∂P and in two or more Voronoi cells, is called a *Voronoi vertex*. Maximal simple curves contained in the union of the cell boundaries and not containing Voronoi vertices are *Voronoi edges*. A maximal subcurve of a Voronoi edge that is a contiguous portion of a single hyperbola or a straight line is referred to as an *arc*. Endpoints of arcs which are not Voronoi vertices are called *breakpoints*. Note that a breakpoint necessarily has degree two, as degree one is excluded by corollary 3.24 and more than three arcs sharing an endpoint would constitute a Voronoi vertex. First, let us consider the graph defined by Voronoi edges and vertices and ignore the individual arcs and breakpoints. By construction, it is a planar graph with vertices of degree three and above. By Euler's formula the complexity of this map is linear in the number of its faces. However, for $s \in S$, $interior(V_P(s))$ is star-shaped around s and, in particular, is connected and contains s, so Voronoi cells correspond one-one to faces of the planar map. Therefore there are only k faces in this map, proving that there are $O(k)$ Voronoi vertices and edges.

Recall that ∂P has n edges, thus adding n vertices and edges to the complexity of the above planar map as soon as we "explode" into constituent segments the portions of ∂P which play the role of the exterior boundary. It remains to estimate the total number of breakpoints on Voronoi edges. We claim that the number of such points is $O(n)$, thus bounding the total complexity of the map in question by $O(n+k)$, and proving that the size of the desired Voronoi diagram is linear in the size of the input. Augment the planar map induced by the Voronoi diagram by the following edges: For each $s \in S$, add to the map segments of $E(P,s)$ truncated to $interior(V_P(s))$. Segments of $E(P,s)$ that do not intersect the interior of $V_P(s)$ are simply discarded. Note that the *only* segments that remain are those emanating from vertices of P in the interior of $V_P(s)$ (Figure 6).
A Voronoi edge that is a portion of bisector $b(s,t)$ in the original map has breakpoints *precisely* at the points of its intersection either with segments of

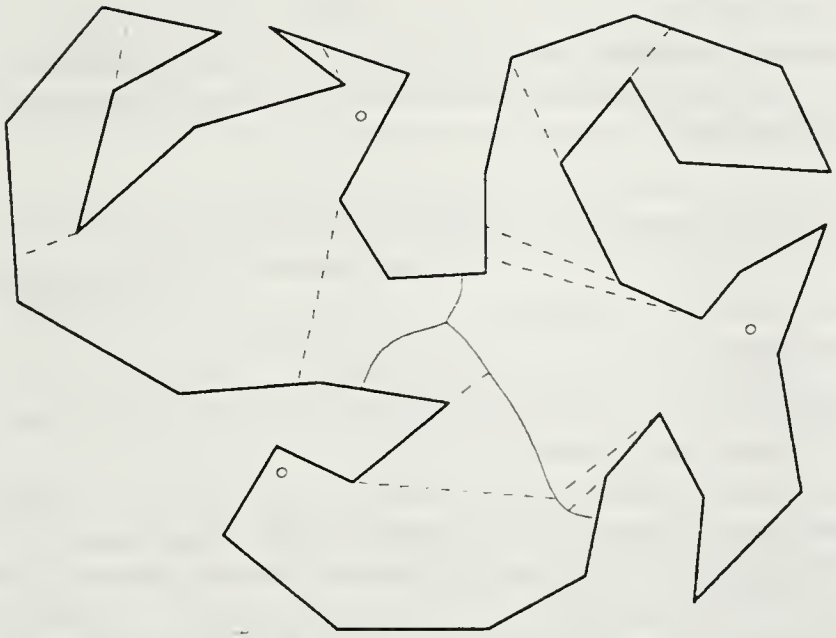


Figure 6.
Augmented geodesic Voronoi diagram of three sites of Figure 1.

$E(P,s)$ coming from $H(s,t)$, or with segments of $E(P,t)$ coming from $H(t,s)$ (lemma 3.22). These are exactly the portions of segments of $E(P,s)$ and $E(P,t)$ that are included in the newly created structure. One should note that by star-shapedness of $interior(V_P(s))$ the intersection of the segment (if any) of $E(P,s)$ emanating from $v \in interior(V_P(s))$ with $interior(V_P(s))$ is indeed a single segment emanating from v. Since such segments cannot intersect the boundary of a Voronoi cell at more than one point by corollary 3.29, and moreover, each segment is in one-one correspondence with the vertex it emanates from, we conclude that the total number of breakpoints is bounded above by the sum of the number of vertices occurring in various Voronoi cell interiors. By corollary 3.26 the interiors of Voronoi cells are disjoint, so the number of breakpoints is bounded above by n, thus ensuring that the sizes of the original map as well as the new (augmented) map are linear in $n+k$. Note that the augmented structure (we will refer to it as the *augmented geodesic Voronoi diagram* $Vor_P^*(S)$) is indeed a planar map, because extension segments are truncated to their respective Voronoi cells and thus are (openly) disjoint from the original map and among themselves. Our algorithm will actually compute the latter map. Thus we obtain:

Theorem 3.30. The complexity of $Vor_P(S)$ is $O(n+k)$ where n is the number of sides of P and $|S| = k$. Moreover, the augmented diagram $Vor_p^*(S)$ also has complexity $O(n+k)$.

4. Finally, the algorithm.

We begin this section with an overview of our algorithm for computing the (augmented) geodesic Voronoi diagram of a set of sites S in a simple polygon P. The steps that require more detailed treatment are discussed at greater length in the following subsections.

Algorithm A.

Input: A simple polygon P of n sides and a set S of k point sites in the interior or on the boundary of P. S and P are assumed to be in general position.

Output: The planar map induced on P by $Vor_P^*(S)$ (cf. Section 3.2).

(1) **Preprocessing**

(i) Triangulate P in $O(n \log\log n)$ time [TV] (or by a simpler $O(n \log n)$ algorithm of [GJPT]).

(ii) Determine for each site in which triangle it lies by constructing the planar map induced by the triangulation, preprocessing it for point-location queries [ST], and performing a query for each site in total $O((n+k)\log n)$ time.

(iii) In $O(n\log n)$ time compute a balanced decomposition of the triangulation tree [Ch] which will allow the main body of our algorithm to recursively cut the polygon into two parts each having at least one quarter of the number of sides in constant time per cut (see also [GHLST]).

(2) **Main part of the algorithm** (recursive)

(i) If S consists of only one site s, $Vor_P(S)$ is a single cell $V_P(s) = P$; $Vor_P^*(S)$ is then computed by utilizing the linear-time shortest path partition construction of [GHLST]; otherwise

(ii) If P is a triangle, compute the Euclidean Voronoi diagram of S in $O(k \log k)$ time, truncate it to P in linear time, recording the intersections of the Voronoi edges with ∂P, and return the result; otherwise

(iii) Split P into two roughly equal polygons P_L and P_R by a cut and divide S into S_L and S_R such that $S_L \subset P_L$ and $S_R \subset P_R$. This can be performed in constant time, as a balanced decomposition of P has been precomputed and the sites are already associated with the triangles in which they lie, thereby making the latter operation implicit so that no processing is required to partition S.

(iv) Recursively compute $Vor_{P_L}^*(S_L)$ and $Vor_{P_R}^*(S_R)$.

(v) Extend $Vor_{P_R}^*(S_R)$ to $Vor_P^*(S_R)$ and $Vor_{P_L}^*(S_L)$ to $Vor_P^*(S_L)$ in time $O((n+k)\log(n+k))$ as described in Section 4.1.

(vi) Compute $Vor_P^*(S)$ by merging $Vor_P^*(S_L)$ and $Vor_P^*(S_R)$ in time $O(n+k)$. The details of this step can be found in Section 4.2.

In part 2 of the algorithm, if either S_L or S_R is empty, omit the corresponding recursive call and skip the merge phase.

As preprocessing takes $O((n+k)\log n)$ time and the recursive part is called on two subproblems of sizes $(\alpha n, k_1)$ and $((1-\alpha)n, k_2)$, where $k_1 + k_2 = k$ and $\frac{1}{4} \leq \alpha \leq \frac{3}{4}$, with the solution obtained from partial solutions, if any, in time $O((n+k)\log(n+k))$, a simple recurrence inequality bounds the running time of the algorithm by $O((n+k)\log(n+k)\log n)$.

4.1. Extending the diagram.

In this section we will describe a procedure implementing step 2(v) of Algorithm A. It computes the augmented geodesic Voronoi diagram for a set of sites in a simple polygon P given the diagram for the same sites in a subpolygon, where the subpolygon was obtained by cutting P along a chord, as in step 2(iii) of Algorithm A.

Let P be a closed region bounded by a simple n-gon. Assume that a triangulation of P is available and let e be a chord of the triangulation. Denote (the closures of) the two polygonal regions into which e cuts P by P_1 and P_2. Suppose S is a set of k sites in P_1. We will describe an algorithm for extending $Vor^*_{P_1}(S)$ to P_2 and thereby obtaining $Vor^*_P(S)$. Extension of the augmented Voronoi diagram proceeds by traversing the triangulation tree of P_2 top-down starting from the triangle adjacent to e. "Visiting" a triangle Δ involves extending the augmented Voronoi diagram to Δ and preparing the structures for further extension to unvisited triangles adjacent to Δ. Construction of the augmented Voronoi diagram in the interior of Δ is performed by a sweeping algorithm. Before we describe the algorithmic details, some properties of the augmented Voronoi diagram of S in P_2 need to be examined. In particular, we will show that the edge structure of $Vor^*_P(S)$ in the interior of P_2 is a forest constructible by sweeping.

Consider the Voronoi diagram of S in P. Let G be the graph formed by Voronoi edges and their incident Voronoi vertices (refer to Figure 7), i.e., the edge and vertex structure of $Vor_P(S)$. Let F be the graph formed by arcs and extension segments restricted to appropriate Voronoi cells together with their incident Voronoi vertices and breakpoints (cf. Section 3.2), i.e., the edge and vertex structure of $Vor^*_P(S)$. Note that G and F are planar. By a slight abuse of notation we identify a graph with its embedding in P so that vertices are identified with points of P and edges—with curvilinear segments in P. In particular, we will write $G \cap P_2$ ($F \cap P_2$) to mean G (respectively, F) restricted to P_2, i.e., G (respectively, F) with vertices outside of P_2 deleted and edges leaving P_2 truncated down to their points of intersection with e.

Lemma 4.1. $G \cap P_2$ is a forest (i.e., a cycle-free undirected graph) with leaves on ∂P_2.

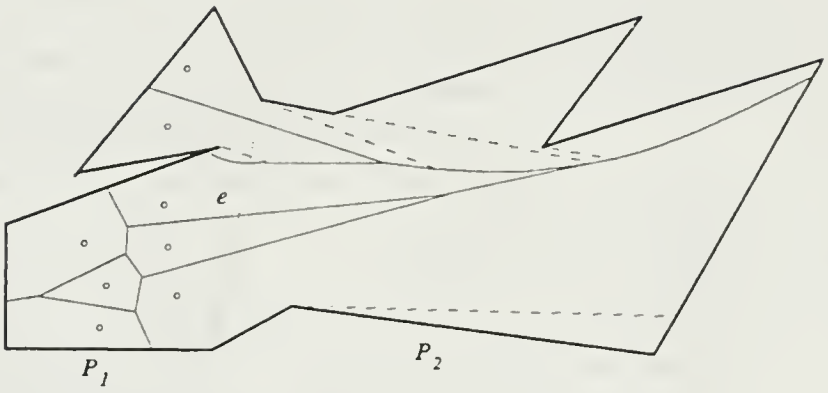


Figure 7. Solid lines represent G, together with dashed $-F$.

Proof. Suppose it contains a non-trivial cycle. Recall that we identify a graph with its embedding in P. Therefore, we may speak of a *minimal* cycle C—one which together with its interior does not properly contain another cycle. As G cannot have vertices of degree one in the interior of P (corollary 3.24), the interior of C contains no point of G, and thus is a single connected component of $P - G$, i.e., a cell of the Voronoi diagram (strictly speaking, the relative interior of a Voronoi cell). Thus there is a Voronoi cell whose interior lies entirely in P_2. However, by corollary 3.25, a Voronoi cell must contain its owner site in its interior, contradicting the assumption $S \subset P_1$. □

Corollary 4.2. $F \cap P_2$ is a forest with leaves on ∂P_2.

Proof. Since, in terms of its embedding in P, F differs from G only by the extension segments truncated to their respective Voronoi cells, it is sufficient to show that extension segments can never close a cycle in G. It is clearly enough to establish that no collection of extension segments[2] contained in one cell can, together with their incident endpoints, form a connected component that meets G twice. Since the extension segments are defined as portions of shortest paths from the owner v of the cell, any two such segments are necessarily disjoint, for otherwise either their common endpoint would have *two* shortest paths to v, or the segments would overlap, which would imply existence of more than one extension segment emanating from a single anchor. It is easily verified that the only other way in which closures of two extension segments can intersect is for one of them to end at the anchor of the other extension segment (refer to Figure 8).
In this situation, it is convenient to consider the extension segment endpoints coinciding on ∂P to be distinct vertices (leaves) of F. Note that this situation can only occur in "degenerate" configurations when either three vertices of P, or two vertices of P and a site of S are collinear. Thus we may consider

[2] From this point on by an "extension segment" we will mean extension segment truncated to the interior of the appropriate Voronoi cell.

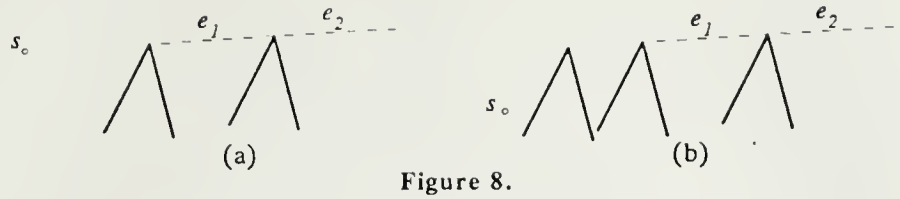


Figure 8.
Degenerate extension segments of $E(P,s)$.
In both cases, e_1 is the degenerate segment.

extension segments effectively disjoint from each other. Now, an extension segment emanates from a vertex v of P and terminates at the next intersection with ∂P or with a Voronoi edge. Under the assumption of general position, v cannot appear on the bisector of two sites and thus on any Voronoi edge. Therefore, an extension segment intersects G at most once, as desired. All leaves of $F \cap P_2$ lie on ∂P_2 since the same is true of leaves of $G \cap P_2$ and extension segments create leaves on ∂P_2 only. □

We now proceed to analyze the structure of the forest $F \cap P_2$ in order to justify the sweeping algorithm for computing it, which is given at the end of this subsection.

Lemma 4.3. Consider a triangle Δ which was entered through its side f in pre-order traversal of the triangulation tree of P_2 from (the triangle incident to) e. Let $x \in interior(\Delta)$ be a point on an edge g of $F \cap P_2$. Then the line p tangent to g at x crosses f.

Proof. The lemma is trivially true for an extension segment g as g follows the shortest path from a site and thus must enter Δ through f—there can be no anchors in the interior of Δ.

Consider an arc g of $b(s,t)$ (see Figure 9).
Since the shortest paths from x to s and t enter Δ through f and there are no anchors in the interior of Δ, $\vec{u}(x,s)$ and $\vec{u}(x,t)$ must be directed in such a way as for the half-lines containing them to cut f. But by lemma 3.22 p bisects the angle between $\vec{u}(x,s)$ and $\vec{u}(x,t)$, so it must also meet f. □

Lemma 4.4. Let Δ and f be as above. Then all edges of $G \cap P_2$ that meet f intersect it transversally.

Proof. Let f' and f'' be the two remaining sides of Δ. Consider an arc g of $b(s,t)$ non-transversally meeting f at x. Note that x is an interior point of f since, on the assumption of general position, $b(s,t)$ cannot contain vertices of P. Since the shortest paths from x to s and to t must not come from f', f'', or the interior of Δ, both $\vec{u}(x,s)$ and $\vec{u}(x,t)$ must lie in the closed half-plane bounded by f and not containing Δ. The only possible way of keeping this consistent with lemma 3.22 that states that the tangent to g at x, i.e., f, must bisect the angle between $\vec{u}(x,s)$ and $\vec{u}(x,t)$ is for the latter two vectors to coincide and point along f. But coincidence of $\vec{u}(x,s)$ and $\vec{u}(x,t)$ on a bisector is impossible in general position by the proof of lemma 3.22.

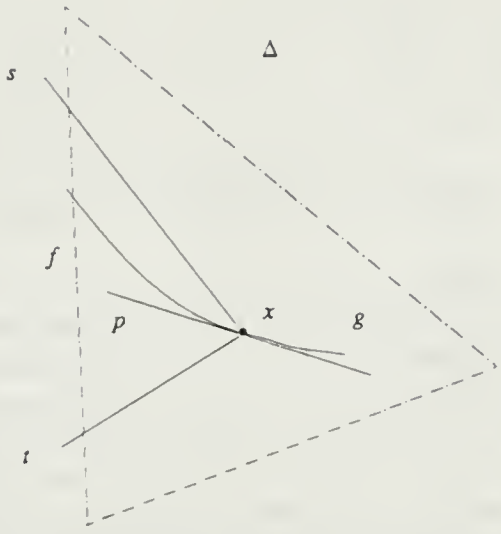


Figure 9.

Contradiction. □

Note 4.5. The obvious extension of the previous lemma to all edges of $F \cap P_2$ does not necessarily hold. The reason is that, though ordinarily extension segments meet f transversally, in some degenerate configurations (cf. Figure 8) an extension segment might coincide with *interior*(f). We will hereafter refer to these as *degenerate extension segments*.

We now proceed to define a "natural" orientation on $F \cap P_2$, formalizing the intuitive notion "away from e", that transforms $F \cap P_2$ into a root-directed forest. Orient edges of $F \cap P_2$ as follows: Again, let Δ be a triangle of P_2 entered through f. Given an edge g, a point x on g in the interior of Δ or of f, the tangent p to g at x intersects f transversally. Consider the direction of traversal of g for which the velocity vector at x points along p *away* from $f \cap p$ (Figure 10a). In case $x \in f$ direct the velocity vector *inward* Δ (Figure 10b).
A degenerate extension segment that overlaps f is oriented so as to *emanate* from its anchor point. We claim that this orientation is well-defined and indeed makes $F \cap P_2$ a root-directed forest. To show this, it is sufficient to prove that the above method indeed assigns a *unique* orientation to each edge independent of the point x selected to define the orientation and that no vertex has out-degree greater than one.

Lemma 4.6. The orientation of an edge g is well-defined, i.e., there is a traversal of g consistent with the above requirements on the direction of the velocity vector at every point of g.

Proof. It is clear that the orientation of a degenerate extension segment cannot be inconsistent as it is defined globally rather than locally. Consider

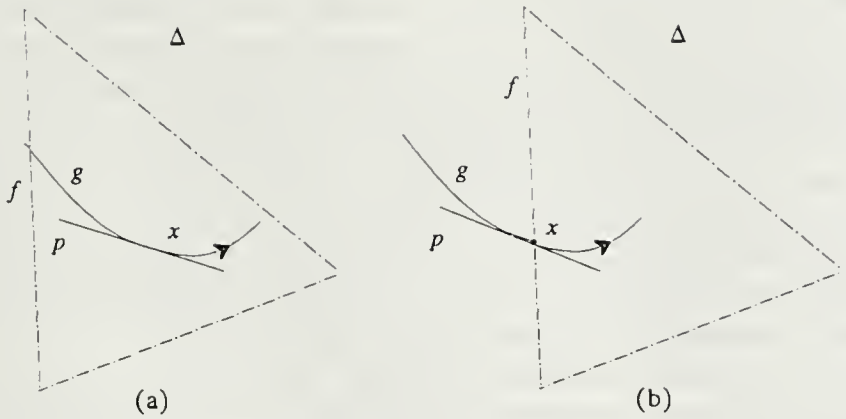


Figure 10. Definition of orientation.

any other type of edge g. Let us start by restricting our attention to a triangle Δ entered through its side f. The orientation is defined in terms of where the tangent to g intersects f. As the tangent varies continuously along g (lemma 3.22) and continues to intersect f (lemma 4.3), it continues to meet f *on the same side* of the point of tangency, so the orientation of g is consistent on a single connected portion of g in the interior of Δ. The ambiguous case of the tangent coinciding with f is excluded by lemma 4.4. Thus it is sufficient to show that the orientation stays consistent when crossing chords separating triangles. And indeed it is, for suppose g crosses the edge f' separating Δ from Δ', i.e., Δ' is entered though f'. By lemma 4.4 g intersects f' transversally. By smoothness of bisectors (lemma 3.22), the tangent to g at $g \cap f'$ still meets f. According to our definition of orientation, g will be oriented outward from Δ in the interior of Δ near f', it will be oriented from Δ to Δ' at its intersection point with f' and it will be oriented inward Δ' in the interior of Δ' near f', thus making the orientation consistent everywhere along g. □

So the orientation of an edge is well-defined. Let us proceed to classify the vertices of the forest $F \cap P_2$ and to prove that the orientation indeed yields a root-directed forest. We observe that the leaves of the directed forest, i.e., vertices with zero in-degree are exactly the points of intersection of F and e together with vertices of ∂P_2 that have extension segments of F emanating from them. These vertices of $F \cap P_2$ indeed have degree one and their incident edges are directed away from them. An internal vertex of $F \cap P_2$ (i.e., neither a root nor a leaf) in the interior of Δ has out-degree of exactly one, as shown by the following lemma. The case of an internal vertex landing on a chord of the triangulation will be handled separately (cf. note 4.9 below).

Lemma 4.7. Every vertex v of $F \cap P_2$ in the interior of Δ has out-degree one.

Proof. Suppose there are two or more edges emanating from v. Note that these edges are *Voronoi edges* and thus v is necessarily a Voronoi vertex, for an extension segment cannot *emanate* from a point in the interior of P. So suppose there are two or more Voronoi edges emanating from v. Choose a pair of *adjacent* edges. As the edges are adjacent, there must be a single Voronoi cell $V_P(t)$ (locally) "wedged" between them and the said edges must be portions of bisectors $b(s,t)$ and $b(t,r)$, for some sites s and r. Let $\hat{t}$ be the anchor of v with respect to t. Note that, by definition of orientation, the tangents l_1 to $b(s,t)$ and l_2 to $b(t,r)$ at v intersect f when extended back through v. By construction, $V_P(t)$ is (locally) wedged between the half-lines of l_1 and l_2 extending *away* from f (see Figure 11a).
In particular, by star-shapedness of $V_P(t)$, t must lie in the wedge W, contradicting the fact that all anchors are contained in the polygon over which the diagram has already been constructed and thus must lie on f or on the side of f opposite Δ and be visible from v through f—in particular, f must lie between v and $\hat{t}$.

Suppose there is *no* edge emanating from v. If v is a breakpoint (i.e., a meeting point of a bisector and an extension segment), two arcs of the bisector join smoothly at v, thus guaranteeing the presence of at least one outgoing edge. Hence v must be a Voronoi vertex. As $G \cap P_2$ has no leaves in the interior of Δ, there must be two or more Voronoi edges incident *to* v. Consider the tangents to these edges at v and choose the two edges whose tangents intersect f closest to either of its two endpoints (cf. Figure 11b). Similarly to the above argument, the larger wedge at v bounded by the two edges must (locally) belong to a single cell $V_P(t)$ for some $t \in S$, and the said edges must be portions of $b(s,t)$ and $b(t,r)$, respectively, for some $s, r \in S$. Let l_1 (l_2) be the line tangent to $b(s,t)$ ($b(t,r)$, respectively) at v. Let W be the smaller wedge at v bounded by the half-lines of l_1 and l_2 that do not intersect f. By star-shapedness of $V_P(t)$, $\vec{u}(v,t)$ must point into W, again contradicting the fact that the path $h(v,t)$ exits Δ through f and its portion

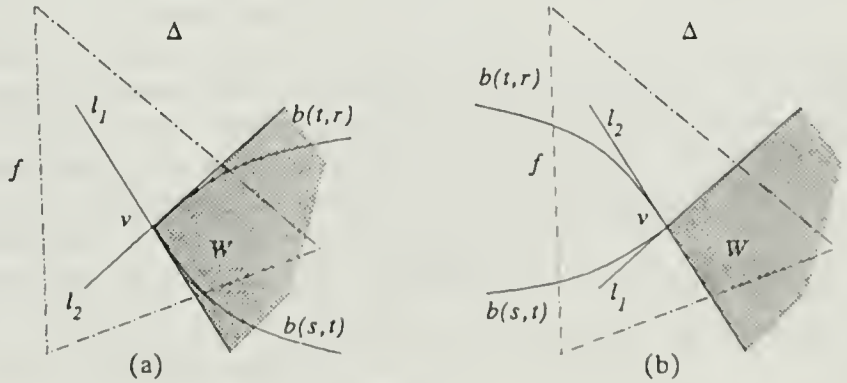


Figure 11.

contained in f is a straight line. □

The vertices of $F \cap P_2$ on ∂P_2 that are neither anchor points nor lie on e are roots as shown by the following lemma.

Lemma 4.8. Let Δ be a triangle entered through f and let its remaining two sides be f' and f''. No vertex v of $F \cap P_2$ in the interior of f' or f'' has an outgoing edge g that emanates from v into the interior of Δ.

Proof. The edge g in question must intersect (say) f' transversally at v (otherwise it is a degenerate extension segment, and so *coincides* with f' by lemma 4.4 and note 4.5) and, by lemma 4.3 and smoothness of arcs, the tangent to g at v must intersect f, making g an *incoming* edge at v by definition of orientation on g. □

Note 4.9. There can exist internal vertices of $F \cap P_2$ that lie on a chord f' separating two triangles Δ and Δ' of P_2 with Δ entered through f and Δ' through f'. In this case, lemma 4.8 shows that there can be no edge leaving v into Δ and reasoning similar to lemma 4.7 shows that there is *exactly* one edge emanating from v into Δ', thereby showing that v has indeed out-degree one and thus is a valid internal vertex of the root-directed forest.

Corollary 4.10. $F \cap P_2$ is a root-directed forest with leaves on e and at anchor points, roots at non-anchor vertices of F on $\partial P_2 - e$, all internal non-root nodes being Voronoi vertices and breakpoints and having in-degree of at least two.

Definition 4.11. Let Δ be a triangle entered through f and let f' and f'' be its two remaining sides. Define the *sweep-curve* s_τ to be the broken line consisting of a segment parallel to f at distance τ from it extending from f' to f'' together with two portions of f' and f'' from the endpoints of f to the endpoints of the said segment. Let r be the distance from f to the vertex of Δ incident to f' and f''.

Corollary 4.12. In this notation, s_τ (for $0<\tau<r$) separates the vertices of $F \cap P_2$ into two sets in such a way that all (directed) edges between two vertices of different sets emanate from below s_τ (i.e., from vertices lying on the side of s_τ that contains f) and terminate above s_τ. Moreover, if τ is such that s_τ passes through a vertex v of $F \cap P_2$ in the interior of Δ or f, there is *precisely one* edge incident *from* v and the said edge extends from v into the area *above* s_τ (Figures 12a&b).

Lemma 4.13. $F \cap \Delta$ $(=(F \cap P_2) \cap \Delta)$ can be constructed by sweeping Δ with s_τ.

Proof. All that is needed to build F in Δ is to systematically locate a pair of children with a common parent in $F \cap \Delta$, and replace the children by the parent, thereby advancing the sweep-curve through the forest one step. This in turn reduces to maintaining the intersection of F with current sweep-curve and repeatedly locating the pair of adjacent edges whose intersection point lies on s_τ with least τ. □

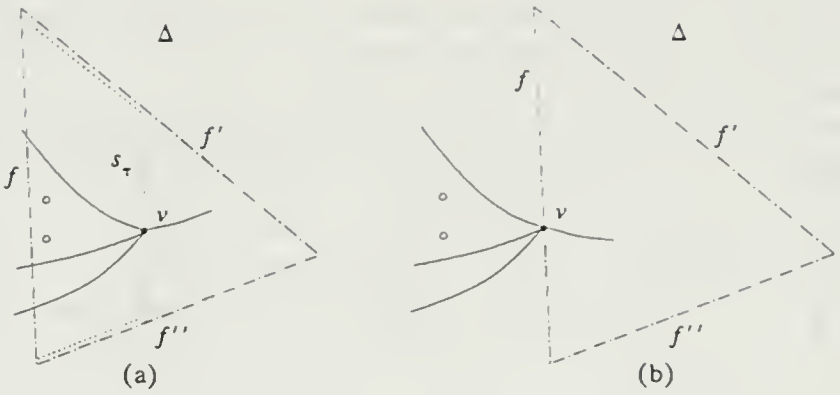


Figure 12. Behavior at a vertex of $F \cap P_2$.

We are now ready to describe the algorithm for extending $Vor^*_{P_1}(S)$ to P_2. Throughout the algorithm $F \cap P_{current}$ for the current polygon $P_{current}$ is maintained while $P_{current}$ grows from P_1 to P. More precisely, maintain the planar map induced by $\partial P_{current}$ together with $F \cap P_{current}$ (i.e., $Vor^*_{P_{current}}(S)$). For each bounded face R of this map (we will refer to it as a *region*), maintain the following information: the site s to whose Voronoi cell R belongs, the anchor $\hat{s}$ for the shortest path from s to an arbitrary point of R, and the value $d(s,\hat{s})$. Therefore, given an arbitrary face R of the planar map and a point x in R, the site s closest to x and the distance $d(s,x)$ can be determined in constant time.

We will assume that the planar map for P_1 is available initially. Since possessing $Vor^*_{P_1}(S)$ implies knowledge of the sorted order in which edges of F meet the edge e separating P_2 from P_1, we can build in linear time a search tree containing this information. More precisely, the tree will represent those regions of the induced planar map on P_1 that are adjacent to e. Namely, it will contain for each such region R the *owner* of R (the site to whose Voronoi cell R belongs), the *anchor* of R (the last vertex on the shortest path from the owner to an arbitrary point in R) and the *weight* of the anchor (the distance between the owner and the anchor). The tree must support *insert*, *delete* and *split* operations in time logarithmic in its maximum size (see, for example, red-black trees of [GS]).

The algorithm for constructing $F \cap P_2$ from $Vor^*_{P_1}(S)$ (and thus $F \cap P_1$) and thereby obtaining F will proceed by performing some initial setup at the edge e and then traversing the triangulation tree of P_2 starting from the triangle adjacent to e:

Algorithm B

Input: A triangulated polygon P separated by a chord e of the triangulation into subpolygons P_1 and P_2, and the planar map induced on P_1 by $Vor^*_{P_1}(S)$.

Output: The planar map induced on P by F and ∂P (i.e., $Vor^*_P(S)$).

(1) *Initial step*.

(i) From $Vor^*_{P_1}(S)$, extract the ordered list L of regions adjacent to e.

(ii) Convert the list into a search tree T.

(iii) For each pair of adjacent edges bounding a single region on L (and thus in T) compute their point of intersection without checking whether the resulting intersections are feasible in any sense. Perform a point location query on each of the points obtained and collect them into buckets corresponding to the triangle where each of those intersection points lands (an equally feasible approach would be to prioritize the buckets immediately).

(2) Process every triangle of the triangulation of P_2 in a pre-order traversal of its triangulation tree, starting at the triangle adjacent to e and treating the tree as rooted at that triangle. The order in which children are visited is irrelevant. Upon entering, through edge f, a triangle Δ, whose remaining edges are f' and f'', whose vertex shared by f' and f'' is v, and whose associated tree T represents the regions of the planar map intersecting f, perform:

(i) Let x be the common endpoint of f and f'. Locate the anchor y of the region to which x belongs. Then consider the ray out of x directed away from y. If this ray partially overlaps f, create a degenerate extension segment that emanates from x and follows f until its first intersection with another edge of F as represented by T. If the said ray intersects the interior of Δ, create an extension segment emanating from x and directed away from y. In either case, add the intersection of the new extension segment with its neighbor in T to the bucket of the appropriate triangle and create a new region sandwiched between the new extension segment and f' with its owner being the owner of y, its anchor being x and the weight of its anchor being the sum of $d(x,y)$ and the weight of y. Otherwise there is no need to create a new extension segment or a new region. Repeat the same procedure for the other endpoint of f. Note that creation of a region involves *insertion* into T.

(ii) Construct a priority queue Q containing all intersections in the bucket associated with Δ. Q is ordered by increasing distance from the line containing f, which amounts to sweeping across Δ with s_τ. (Let us note that a different ordering would work just as well as long as on every straight segment in Δ intersecting f the relation reduces to the total order along the segment.

(iii) The intersection p in Q closest to f is then repeatedly deleted from Q. If p is actually valid, i.e., it refers to an intersection of two curves currently adjacent on s_τ (i.e., bounding a common region in T), it

must correspond to the disappearance of the region between them from T and the replacement of the two bisector arcs (or a bisector arc and an extension segment) by a new bisector arc. At this point the region is deleted from T. If p does not represent a valid intersection, it is simply discarded. The two newly created pairs of curves adjacent on s_τ have their intersections computed and located in the triangulation. If they land in Δ, they are added to Q; otherwise, they are added to the appropriate buckets. The process repeats until the queue is empty.

(iv) When the construction inside Δ is completed, the two trees corresponding to f' and f'' are produced by splitting the current version of T, (conceptually) cutting s_τ at v. Note that s_τ coincides with $f' \cup f''$ at this point, so splitting T correctly creates the two initial trees representing s_0 for the two unvisited triangles adjacent to Δ. In case f' (respectively f'') is actually an edge of P, the corresponding search tree is converted to adjacency information for F along it.

We proceed to analyze the time complexity of Algorithm B. Steps 1(i) and 1(ii) clearly take linear time. As there is only a linear number of initial candidate pairs of adjacent curves on s_τ, step 1(iii) requires $O((n+k)\log n)$ time. One can easily see that the time complexity of part 2 is $O(\log n)$ per candidate intersection for point location, $O(\log(n+k))$ for queue maintenance operations (as no candidate is ever in two queues), $O(\log(n+k))$ for tree deletion for each *valid* intersection (i.e., a Voronoi vertex or a breakpoint), and $O(\log(n+k))$ for tree insertion and split per vertex of P_2. The last three bounds follow as soon as it is shown that the size of all queues and trees is at all times $O(n+k)$. In fact, each candidate intersection is either created initially (there are only $O(|L|)=O(n+k)$ of such) or added during a deletion from or insertion into a tree (at most two new candidates per insertion, one new candidate per deletion). There are at most n insertions and $O(n+k)$ deletions—note that the universe does not necessarily decrease all the time; in fact it can grow by as much as n due to introduction of new extension segments, which can occur at most once per vertex of P_2. Thus the *total* size of all queues and trees is at every point of the algorithm bounded by $O(n+k)$, as desired. In other words, part 2 spends logarithmic time per candidate intersection which, there being $O(n+k)$ candidate intersections, bounds the execution time of part 2 by $O((n+k)\log(n+k))$.

Therefore the extension of the augmented Voronoi diagram of a set of k points from a portion of an n-gon to the whole polygon is accomplished in $O((n+k)\log(n+k))$ time by Algorithm B, providing the desired bound for part 2(v) of Algorithm A. Observe that it is impossible to improve Algorithm B so that it runs in time $O(n+k)$ without modifying the remaining parts of Algorithm A, for such an improvement would reduce the time complexity of the entire algorithm to $O((n+k)\log n)$, contradicting the lower

bound discussed in Section 6.

4.2. Merging the two diagrams.

In this section we describe a linear-time algorithm for merging the Voronoi diagrams of two sets of sites in a polygon assuming that a chord of the polygon separates the two sets of sites. Again, before detailing the algorithm, we will examine some properties of Voronoi diagrams of such sets of sites.

Definition 4.14. Let A be a set of sites in P and x be a point in P. Then the *distance* from x to A is defined by $d(x,A) = \min_{s \in A} d(x,s)$. Let a *shortest path* from x to A be $h(x,A) = h(x,s)$ where $s \in A$ is such that $d(x,s) = d(x,A)$. Note that $h(x,A)$ is not uniquely defined for x on an edge of $Vor_P(A)$.

Definition 4.15. Let $A, B \subset P$ be disjoint non-empty sets of sites. The *bisector* of A and B is $b(A,B) = \{x \in P \mid d(x,A) = d(x,B)\}$.

Note 4.16. It is easily checked that $b(A,B)$ is the union of the Voronoi edges of $Vor_P(A \cup B)$ that separate the cell of an A-site from that of a B-site and the Voronoi vertices that lie on the boundary of at least one cell of a site in A and at least one cell of a site in B. In particular, $b(A,B)$ has complexity linear in the sum of sizes of A, B, and P since it is (the union of) a collection of Voronoi edges and vertices. Moreover, for any Voronoi edge e that $b(A,B)$ contains, it also contains the two Voronoi vertices incident to e. Hence $b(A,B)$ is a subgraph of G (as defined in Section 4.1). Moreover, $b(A,B)$ cannot terminate at Voronoi vertices in the interior of P or contain isolated Voronoi vertices, as the fact that a Voronoi vertex v belongs to $b(A,B)$ already implies that v is in Voronoi cells of at least one A-site and at least one B-site. This in turn implies the existence of at least two (and at least one for $v \in \partial P$) Voronoi edges emanating from v that separate cells of sites from different sets. In fact, this argument shows that there is always an even number of edges belonging to $b(A,B)$ and emanating from an arbitrary Voronoi vertex in the interior of P. Thus $b(A,B)$ can be decomposed into edge-disjoint simple paths and cycles. Paths connect two points of ∂P and cycles can be considered not to touch ∂P, for a cycle that does touch it can be regarded as a path connecting a point of ∂P to itself.

Definition 4.17. $H(A,B) = \{x \in P \mid d(x,A) < d(x,B)\}$. In other words, $H(A,B)$ is the set of points in P lying closer to some site of A than to any site of B.

Notice that trivially $\{H(A,B), b(A,B), H(B,A)\}$ is a partition of P and, by a simple continuity argument, $b(A,B)$ actually separates $H(A,B)$ from $H(B,A)$ in the sense that they cannot be connected by a path that does not meet $b(A,B)$.

Lemma 4.18. Suppose $s \in A$. Let $V_P(s)$ denote the Voronoi cell of s in $Vor_P(A)$ and $V'_P(s)$ denote its cell in $Vor_P(A \cup B)$. Then

$$V'_P(s) = V_P(s) \cap (H(A,B) \cup b(A,B)).$$

Proof.

$$V'_P(s) = \{x \mid d(s,x) = \min_{r \in A \cup B} d(r,x)\}$$

$$= \{x \mid d(s,x) = \min_{r \in A} d(r,x)\} \cap \{x \mid \min_{r \in A} d(r,x) \leq \min_{r \in B} d(r,x)\}$$

$$= V_P(s) \cap \{x \mid d(x,A) \leq d(x,B)\} = V_P(s) \cap (H(A,B) \cup b(A,B)).\square$$

Similarly one shows that $interior(V'_P(s)) = interior(V_P(s)) \cap H(A,B)$. In particular, since $interior(V'_P(s))$ is connected, it is simply the connected component of the set $interior(V_P(s)) - b(A,B)$ containing s.

Lemma 4.19. Suppose that $Vor^*_P(A)$ and $Vor^*_P(B)$ are given. Then, given a point on every connected component of $b(A,B)$, both $b(A,B)$ and $Vor^*_P(A \cup B)$ can be constructed in time $O(|A|+|B|+n)$ where n is the number of vertices of P.

Proof. From each point given on $b(A,B)$ build connected components of $b(A,B)$ by the usual Shamos-Hoey scan of the two Voronoi diagrams (see, for example, [L] or [SH]). Note that one can follow a bisector of two sites as long as the two anchor points stay fixed. The anchor points change precisely when the bisector crosses an extension segment, thereby moving from one region of the shortest path partition around a site to another. At this point one can recover the new anchor point as it is simply the anchor associated with the shortest path partition region being entered. The fine structure of the Voronoi cells (namely their shortest path partition) is somewhat similar to the "spokes" of Kirkpatrick [K]. A complication arises when vertices of degree greater than two are encountered on $b(A,B)$. At such a Voronoi vertex v the list of owners and anchors of adjacent Voronoi cells (and shortest path partition regions) is known, allowing one to determine *all* the edges leaving v that separate a cell of an A-site from that of a B-site, thus enabling the Shamos-Hoey scan to trace *all* branches of $b(A,B)$. The additional effort required is proportional to degree of v in $Vor_P(A \cup B)$, thus it is linear over the construction of $b(A,B)$. Note that this complication could be avoided completely by taking "general position" to mean, in addition, that no four points are on a geodesic circle, thus excluding Voronoi vertices of degree above three and forcing degree of at most two for $b(A,B)$.

Now split both augmented Voronoi diagrams along $b(A,B)$ and collect the portions of Voronoi cells reachable from their owners; the resulting collection is (by lemma 4.18) the augmented Voronoi diagram of $A \cup B$. Note that one need not be concerned with updating the shortest path partition of the cells, as the effect of the merge on it is limited to truncation of extension segments down to new cell boundaries, which is done automatically during the Shamos-Hoey scan. □

Observe that the two steps described above need not be done separately. It is more convenient to perform truncation of Voronoi cells by $b(A,B)$ at the time it is being constructed, concurrently updating the two Voronoi diagrams and merging them along $b(A,B)$.

Note 4.20. To repeatedly locate the intersection of a (simple) arc of a bisector with the boundary of "current" regions of $Vor_P^*(S_1)$ and $Vor_P^*(S_2)$ in an efficient manner, we follow Kirkpatrick [K] and introduce a further refinement of our map that will guarantee that each individual region has bounded complexity, thereby allowing us to trace an arc of the bisector within each such subregion in constant time. In this extra refinement, in addition to the extension segments, every vertex of ∂P and every Voronoi vertex lying in a Voronoi cell are connected by a shortest path to the owner of the cell (actually, one needs only connect each such point to its anchor by a straight segment, so the resulting structure is still linear). The structure described coincides with the "Voronoi triangulation" of [AA] (Figure 13). Moreover, since each segment introduced is contained in the shortest path to the owner, it will not intersect the "contour" $b(A,B)$ more than once. It is also easy to see that every face of the resulting map is bounded by two segments and (in general) an arc of a hyperbola or a polygon boundary segment, and thus has complexity bounded by a constant. It is easy to verify that this finer structure allows Kirkpatrick's tracing procedure to run in linear time.

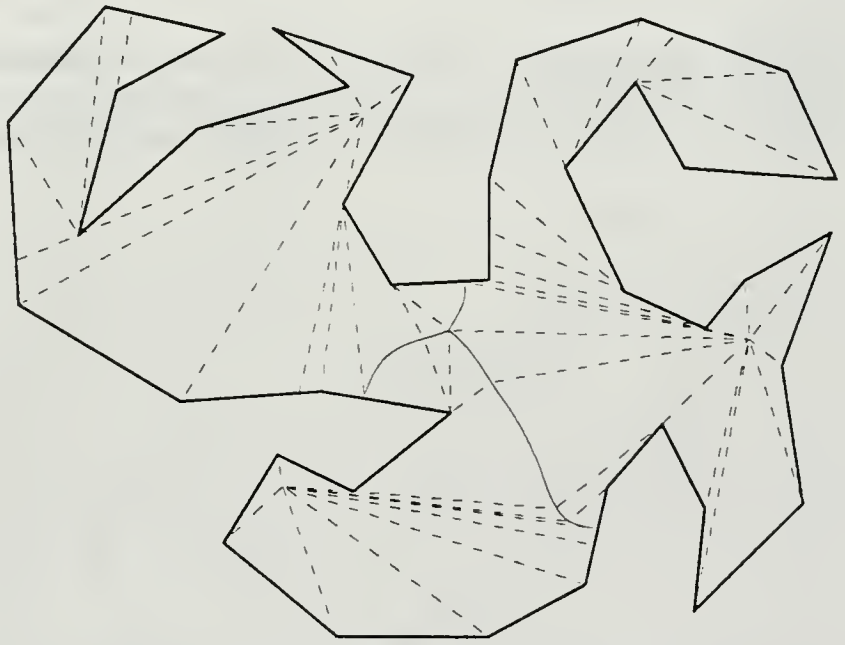


Figure 13.
Voronoi triangulation of P for three sites of Figure 1.

Corollary 4.21. If all components of $b(A,B)$ meet ∂P, $Vor_P^*(A \cup B)$ can be produced from $Vor_P^*(A)$ and $Vor_P^*(B)$ in time $O(|A|+|B|+n)$.

Proof. It is sufficient to locate all points of $b(A,B) \cap \partial P$. Since both $Vor_P^*(A)$ and $Vor_P^*(B)$ are linear in size, it is possible to trace ∂P, in linear time partitioning it into $O(n+k)$ straight segments each of which lies in exactly one region of each partition. Each segment contains points not only nearest to a unique $s \in A$ and a unique $t \in B$ but also having a unique pair of anchors $(\hat{s},\hat{t})$ with $d(s,\hat{s})$ and $d(t,\hat{t})$ known. Thus, on each segment, the analytic expressions for

$$d(x,A) = d(x,s) = |x - \hat{s}| + d(\hat{s},s) \text{ and}$$
$$d(x,B) = d(x,t) = |x - \hat{t}| + d(\hat{t},t)$$

are known. Since the two functions are well-behaved (one would like to say "polynomial of degree two", but they in fact involve radicals), the points where they coincide are bounded in number by a constant (actually, by two) and can be computed in constant time. Each such point corresponds to an intersection of $b(A,B)$ and ∂P, thereby enabling us to compute *all* such intersections in linear time and permitting the application of lemma 4.19.

Note that the case when the functions $d(x,A)$ and $d(x,B)$ coincide is easily shown to violate the general position assumption as in such a case $\hat{s}=\hat{t}$. □

Lemma 4.22. If e is an internal chord of P and A and B are sets of sites of P on opposite sides of e, $b(A,B)$ contains no cycles. In particular, the conditions of corollary 4.21 are satisfied.

Proof. Denote the part of P that contains A by P_1 and the complementary part by P_2 (refer to Figure 14).
Notice that $H(A,B)$ is the union of interiors of Voronoi cells of A-sites (in

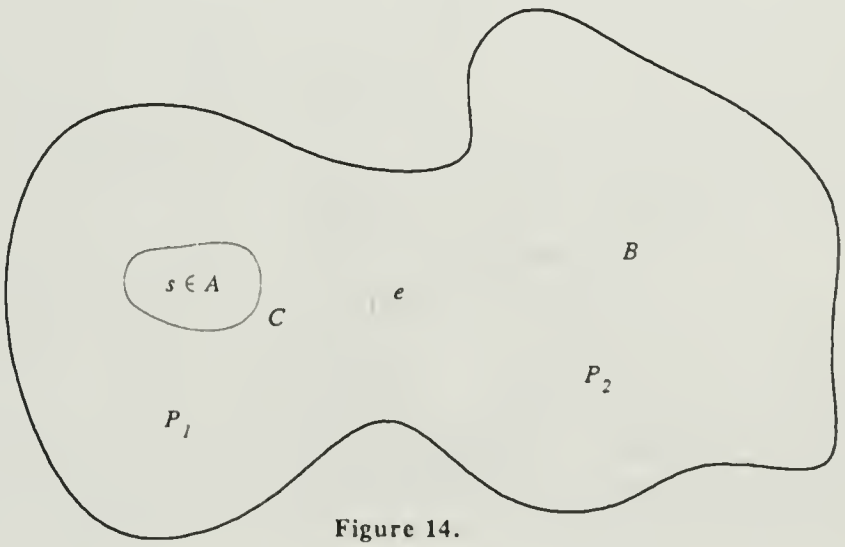


Figure 14.

$Vor_P(A \cup B)$) together with the Voronoi edges separating these cells. Hence, if there is a cycle C in $b(A,B)$, it must completely enclose a cell of, say, A-site s, i.e., there is no path in $H(A,B)$ connecting s to ∂P. Choose x to be the point of e closest to s. Observe that either x lies in the interior of e and $\vec{u}(x,s)$ is perpendicular to e, or x is one of the endpoints of e and the angle between $\vec{u}(x,s)$ and the vector pointing along e away from x has measure $\geq \pi/2$. Extend $h(x,s)$ past s until it intersects ∂P at point y (Figure 15).
We claim that $h(s,y) \subset H(A,B)$, contrary to our assumption that s is enclosed by a cycle $C \subset b(A,B)$. Consider $z \in h(s,y)$ and $t \in B \subset P_2$. If $x \in h(z,t)$, $d(z,t) => d(z,x) => d(z,s)$ (note that $s \neq t$, since A and B are disjoint), so let us assume $x \notin h(z,t)$. By the above observation, the angle between $\vec{u}(x,z)$ $(= \vec{u}(x,s))$ and $\vec{u}(x,t)$ has measure $\geq \pi/2$. It was shown by [PSR] that the side of a geodesic triangle lying against such an angle is strictly the longest. In particular,

$$d(z,t) > d(z,x) \geq d(z,s).$$

Hence $d(z,B) > d(z,s)$, and therefore $z \in H(A,B)$, as asserted. □

Corollary 4.23. The diagrams $Vor_P^*(S_L)$ and $Vor_P^*(S_R)$ in step 2(vi) of Algorithm A can be merged in time $O(n+k)$.

To summarize, the procedure for merging the two diagrams consists of locating all points of $b(A,B)$ on ∂P by a linear scan followed by tracing of every connected component of $b(A,B)$ *a la* Kirkpatrick [K].

5. A special case.

It is interesting to discuss a variation of our algorithm that computes the geodesic Voronoi diagram for a set of k sites among which are *all* reflex vertices of the enclosing n-gon in $O((n+k)\log(n+k))$ time. This variant

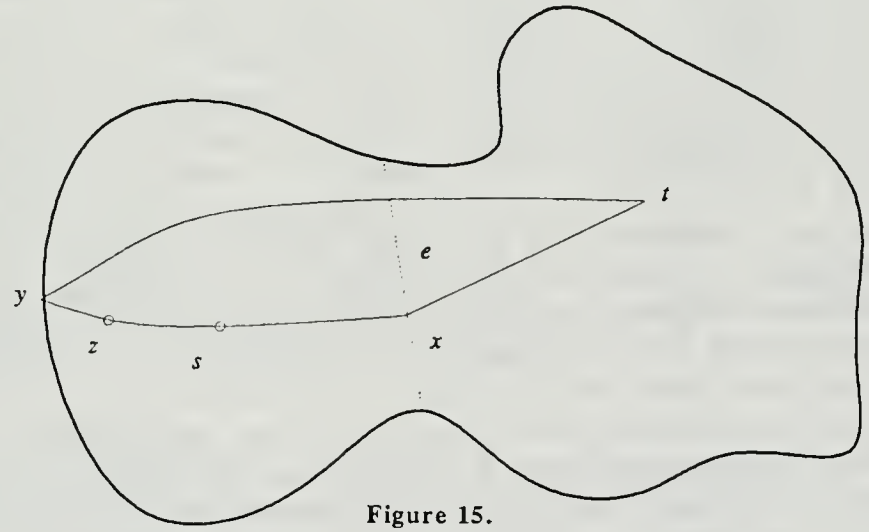


Figure 15.

exploits the following property of the geodesic Voronoi diagram of such a set of sites—the Voronoi cells are star-shaped in the Euclidean metric, as a bend on a path from a point in a cell to the owner of the cell would produce a reflex vertex, i.e., a closer site. The general strategy is still divide-and-conquer, with the polygon recursively split into two roughly equal parts and the resulting diagrams merged *a la* Shamos and Hoey [SH]. However, the extension step is avoided, rather a structure similar to the extension of Voronoi diagram into "site-less" regions is built at the bottom of recursion and maintained together with the Voronoi diagram proper through the merging steps. The rationale behind this approach is that the only sites that can own a given point of the polygon are those visible from it. Consequently, during the merge step, the only points of one half of the polygon that can be "taken over" by sites of the other half are those visible from the sites through the "window" of the dividing chord. In particular, it is enough to construct the Voronoi diagram of the set of sites of one half of the polygon as visible through this window. This notion is identical to the notion of the "peeper's Voronoi diagram" of [BS], but fortunately in our case it is easily seen to have linear size. Moreover, we can show that the geodesic Voronoi diagram of a set of sites in a polygon with "peeper's" Voronoi diagrams attached to some of the polygon edges (in fact, to those which are chords of triangulation of the original polygon) is a structure linear in the size of the polygon and the number of sites and two such structures can be merged in linear time. This yields an $O((n+k)\log(n+k))$ algorithm for constructing the geodesic Voronoi diagram in this special case. The details of this faster algorithm will be reported separately.

Note that the algorithm sketched above allows, in particular, to compute the geodesic Voronoi diagram of the *vertices* of a simple n-gon in time $O(n\log n)$. An alternative $O(n\log n)$ algorithm is provided by work of Lee and Lin [LL] (see also [Chew] and [WS]), which involves the calculation of the *generalized Delaunay triangulation* of a simple polygon P. It is a triangulation of the polygon with the property that the circle circumscribed around each face of it does not contain (in its interior) any vertex of P *visible* simultaneously from all three vertices of the said triangular face. In the case of a convex polygon, this definition gives precisely the conventional Delaunay triangulation of P, which is the dual of the (conventional) Voronoi diagram of the vertices of P, which in turn coincides (inside of P) with the geodesic Voronoi diagram of the vertices of P. In fact, it is not difficult to observe that, for a general simple polygon P, the generalized Delaunay triangulation is (essentially) the dual of the geodesic Voronoi diagram (in fact, it can be shown to have more edges than the dual). [LL] (and also [Chew], [WS]) provide an $O(n\log n)$ algorithm for constructing this triangulation, and the diagram can be reconstructed from it in linear time by visiting the neighbors of each vertex in cyclical order and thereby reconstructing its Voronoi cell. Thus we obtain an alternative $O(n\log n)$ algorithm for computing the geodesic

Voronoi diagram of a set of vertices of a simple polygon (with respect to that polygon).

6. Open problems.

It remains an open problem to determine whether the geodesic Voronoi diagram of a general set of k sites in an n-gon can be computed in time $O((n+k)\log(n+k))$ or, for that matter, any faster than $\Omega((n+k)\log(n+k)\log n)$. Let us observe that the Euclidean closest-pair problem is linear-time reducible to the geodesic Voronoi diagram problem as follows: Given a set S of k points, construct in linear time a rectangular box B large enough to enclose the convex hull of S. At this point, computing $Vor_B(S)$ produces the (conventional) Voronoi diagram of S truncated to B. However, it has been shown (see, for example, [PrS]) that the shortest Euclidean distance between two points of S is realized by a pair of points whose cells are adjacent and the segment connecting which intersects the edge common to the two cells—hence the intersection point must lie in the convex hull of S and thus in B. In particular, we can locate the closest pair in linear time given the truncated Voronoi diagram, which shows an $\Omega(k\log k)$ lower bound for computing the geodesic Voronoi diagram. Recalling the $\Theta(n+k)$ bound on the complexity of the geodesic Voronoi diagram, we obtain the best currently known lower bound of $\Omega(n+k\log k)$.

There are a number of other open problems quite closely related to geodesic Voronoi diagrams. One of them is removing the general position assumptions and defining a cell of a *set* of sites to contain all points of P to which all sites of the set are simultaneously closest (see, for example, [ES]). The objective, once again, would be to estimate the complexity of this diagram and devise an efficient algorithm for constructing it. Another possible avenue for extending our results is removing the restriction that sites be points. Can our approach be extended to yield an efficient algorithm for computing the diagram of a set of straight-line segments in a polygon under a suitable metric, e.g., using an appropriate modification of Yap's technique [Ya]? It might also be interesting to consider which properties of the geodesic distance (such as those discussed in Section 3) survive if we allow the sides of P to be curvilinear.

A somewhat unrelated question that can be raised in conjunction with a geodesic Voronoi diagram is whether the sweeping approach of Fortune [F] can be generalized for constructing such diagrams. A significant obstacle to such a generalization is the absence of an obvious natural equivalent of a straight sweep-line.

The natural problem of analyzing the *furthest-point* geodesic Voronoi diagram is considered in [AFW]. They show a $\Theta(n+k)$ bound on the complexity of the diagram and produce an $O((n+k)\log(n+k))$ algorithm for constructing it.

7. Acknowledgements.

I wish to thank my Ph.D. advisor Micha Sharir for suggesting this problem, for valuable discussions on its solution, and for having the patience to read through numerous versions of this paper. I would especially like to thank the anonymous referee for his/her constructive comments and suggestions.

8. Bibliography.

[A] F. Aurenhammer, "Power diagrams: properties, algorithms and applications," *SIAM J. Computing*, **16** (1987), 78-96.

[AA] T. Asano and T. Asano, "Voronoi diagram for points in a simple polygon," *Discrete Algorithms and Complexity: Proc. Japan-US Joint Seminar,* (Perspectives in Computing, Vol. 15), Academic Press, 1987, 51-64.

[AE] F. Aurenhammer and H. Edelsbrunner, "An optimal algorithm for constructing the weighted Voronoi diagram in the plane," *Pattern recognition*, **17** (1984), 251-257.

[AFW] B. Aronov, S. Fortune, and G. Wilfong, "The furthest-site geodesic Voronoi diagram," *Proc. 4th ACM Symp. on Computational Geometry,* 1988, pp. 229-240.

[BS] A. Baltsan and M. Sharir, "On the shortest paths between two convex polyhedra," *J. ACM*, **35** (1988), 267-287.

[Ch] B. Chazelle "A theorem on polygon cutting with applications," *Proc. 23rd Symp. on Theory of Computing,* 1982, pp. 339-349.

[Chew] L.P. Chew, "Constrained Delaunay triangulations," *Algorithmica*, to appear.

[CD] L.P. Chew and R.L. Drysdale, III, "Voronoi diagrams based on convex distance functions," *Proc. of the ACM Symp. on Computational Geometry*, 1985, pp. 235-244.

[ES] H. Edelsbrunner and R. Seidel, "Voronoi diagrams and arrangements," *Discrete and Comput. Geom.*, **1** (1986), 25-44.

[F] S.J. Fortune, "A sweepline algorithm for Voronoi diagrams," *Algorithmica*, **2** (1987), 153-174.

[GHLST] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan, "Linear time algorithms for visibility and shortest path problems inside a triangulated simple polygon" *Algorithmica*, **2** (1987), 209-233.

[GJPT] M.R. Garey, D.S. Johnson, F.P. Preparata, and R.E. Tarjan, "Triangulating a simple polygon," *Information Processing Letters*, **7** (1978), 175-179.

[GS] L.J. Guibas and R. Sedgewick, "A dichromatic framework for balanced trees," *Procs. 19th IEEE Symp. on Foundations of Computer Science*, 1978, pp. 8-21.

[IIM] H. Imai, M. Iri, and K. Murota, "Voronoi diagram in the Laguerre geometry and its applications," *SIAM J. Computing*, **14** (1985), 93-105.

[K] D.G. Kirkpatrick, "Efficient computation of continuous skeletons," *Procs. 20th IEEE Symp. on Foundations of Computer Science*, 1979, pp. 18-27.

[L] D.T. Lee, "Proximity and reachability in the plane," PhD dissertation, Tech. Report No. R-831, Coordinated Science Laboratory, University of Illinois at Urbana, 1978.

[L2] D.T. Lee, "Two dimensional Voronoi diagrams in the L_p-metric," *J. ACM*, **27** (1980), 604-618.

[LL] D.T. Lee and A.K. Lin, "Generalized Delaunay triangulations for planar graphs, " *Discrete and Comput. Geom.*, **1** (1986), 201-217.

[Lea] W. Lenhart, R. Pollack, J. Sack, R. Seidel, M. Sharir, S. Suri, G. Toussaint, S. Whitesides, and C.K. Yap, "Computing the link center of a simple polygon," *Discrete Comput. Geom.*, **3**(1988), 281-293.

[LP] D.T. Lee and F.P. Preparata, (1984) "Euclidean shortest paths in the presence of rectilinear barriers," *Networks*, **14** (3), 393-410.

[LS] D. Leven and M. Sharir, "Intersection and proximity problems and Voronoi diagrams," in *Algorithmic and geometric aspects of robotics*, J.T. Schwartz and C.K. Yap, eds., (Advances in robotics, Vol. 1), Erlbaum Associates, Hillsdale, NJ, 1987, 187-228.

[LW] D.T. Lee, and C.K. Wong, "Voronoi diagrams in L_1-(L_∞-)metrics with 2-dimensional applications," *SIAM J. Computing*, **9** (1980), 200-211.

[OSY1] C. Ó'Dúnlaing, M. Sharir, and C.K. Yap, "Generalized Voronoi diagrams for moving a ladder: I. Topological analysis," *Comm. Pure Applied Math.*, **39** (1986), 423-483.

[OSY2] C. Ó'Dúnlaing, M. Sharir, and C.K. Yap, "Generalized Voronoi diagrams for moving a ladder: II. Efficient construction of the diagram," *Algorithmica*, **2** (1987), 27-59.

[PSR] R. Pollack, M. Sharir, and G. Rote, "Computing the geodesic center of a simple polygon", *Discrete and Comput. Geom.*, to appear.

[PrS] F.P. Preparata and M.I. Shamos, *Computational Geometry: An introduction*, Springer, New York, 1985.

[SH] M.I. Shamos and D. Hoey, "Closest-point problems," *Procs. 16th IEEE Symp. on Foundations of Computer Science*, 1975, pp. 151-162.

[S] S. Suri, "Computing the link diameter of a simple polygon," Tech. Report JHU/EECS-86/09, Dept. of Elec. Eng. and Comp. Sci., Johns Hopkins University, 1986.

[S2] S. Suri, "Computing the geodesic diameter of a simple polygon," Tech. Report JHU/EECS-86/08, Dept. of Elec. Eng. and Comp. Sci., Johns Hopkins University, 1986.

[S3] S. Suri, "The all-geodesic-furthest neighbor problem for simple polygons," *Procs. 3rd ACM Symp. on Computational Geometry*, June 1987, pp. 64-75.

[ST] N. Sarnak and R.E. Tarjan, "Planar point location using persistent search trees", *CACM*, **29** (1986), 669-679.

[T] G. Toussaint, "An optimal algorithm for computing the relative convex hull of a set of points in a polygon," *Signal Processing III: Theories and Applications, Proc. of EUSIPCO-86*, North-Holland, 1986, pp. 853-856.

[TV] R.E. Tarjan and C. Van Wyk, "An $O(n \log\log n)$ time algorithm for triangulating a simple polygon," *SIAM J. Computing*, **17** (1988), 143-177.

[WS] C. Wang and L. Schubert, "An optimal algorithm for constructing the Delaunay triangulation of a set of line segments," *Procs. 3rd ACM Symp. on Computational Geometry*, June 1987, pp. 223-232.

[Ya] C.K. Yap, "A $O(n\log n)$ algorithm for the Voronoi diagram of a set of simple curve segments," *Discrete and Computational Geometry*, **2** (1987), 365-394.

NYU COMPSCI TR-390 c.2
Aronov, Boris
On the geodesic Voronoi
diagram of point sites in
a simple polygon.

DATE DUE	BORROWER'S NAME